FLEXIBLE ALL-DIGITAL RECEIVER FOR BANDWIDTH EFFICIENT MODULATIONS

Andrew Gray, Meera Srinivasan, Marvin Simon, Tsun-Yee Yan
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

ABSTRACT

An all-digital high data rate parallel receiver architecture developed jointly by Goddard Space Flight Center and the Jet Propulsion Laboratory is presented. This receiver utilizes only a small number of high speed components along with a majority of lower speed components operating in a parallel frequency domain structure implementable in CMOS, and can currently process up to 600 Mbps with standard QPSK modulation. Performance results for this receiver for bandwidth efficient QPSK modulation schemes such as square-root raised cosine pulse shaped QPSK and Feher's patented QPSK are presented, demonstrating the flexibility of the receiver architecture.

KEY WORDS

Bandwidth efficient modulation, Feher's quadrature phase shift keying, trellis-coded modulation, frequency domain receiver

INTRODUCTION

Due to demands for rapidly increasing downlink data rates between spacecraft and ground stations, NASA has developed an all-digital variable data rate receiver implemented on a single CMOS ASIC that is capable of processing data rates in excess of 300 Megasymbols per second or 600 Megabits per second using QPSK modulation. Developed jointly by Goddard Space Flight Center and the Jet Propulsion Laboratory, the all-digital parallel receiver (APRX) uses patent pending parallel processing algorithms to perform the functions of demodulation to baseband, detection filtering, and carrier and symbol timing recovery. In order to process high data rates in relatively inexpensive CMOS, these parallel algorithms allow the demodulator to operate at a processing speed that is one-fourth the data rate [1, 2]. The receiver was originally developed to demodulate BPSK and many variations of QPSK, all with standard non-return-to-zero (NRZ) rectangular pulses, with flexibility designed into the parallel algorithms and ASIC implementation in order to expand the receiver's capabilities to the demodulation of more complicated or higher order modulations such as M-ary phase shift keying (MPSK) and quadrature amplitude modulation (QAM).

This paper expands upon previous work by presenting APRX performance for QPSK with spectrally efficient pulse shapes, specifically square-root raised cosine (SRRC) shaped QPSK and Feher's patented QPSK (FQPSK) modulation. Compared to multilevel modulations such as MPSK and QAM, pulse-shaped quadrature modulations achieve spectral containment while preserving a relatively simple receiver structure, specifically in terms of the design of carrier phase and symbol synchronization loops. On the other hand, pulse shaping may introduce inter-symbol interference that results in performance losses unless some type of equalization is used, as is the case with the APRX. In this paper, we present an overview of the APRX architecture, and explain how it is used to demodulate SRRC shaped QPSK and FQPSK signals, followed by software simulation results describing receiver performance in terms of bit error probabilities.

APRX ARCHITECTURE OVERVIEW

Prior to entering the digital receiver, an intermediate stage downconverts the RF data signal to an intermediate frequency (IF) appropriate for A/D conversion. A bandpass filter is used to reject noise and limit the data bandwidth
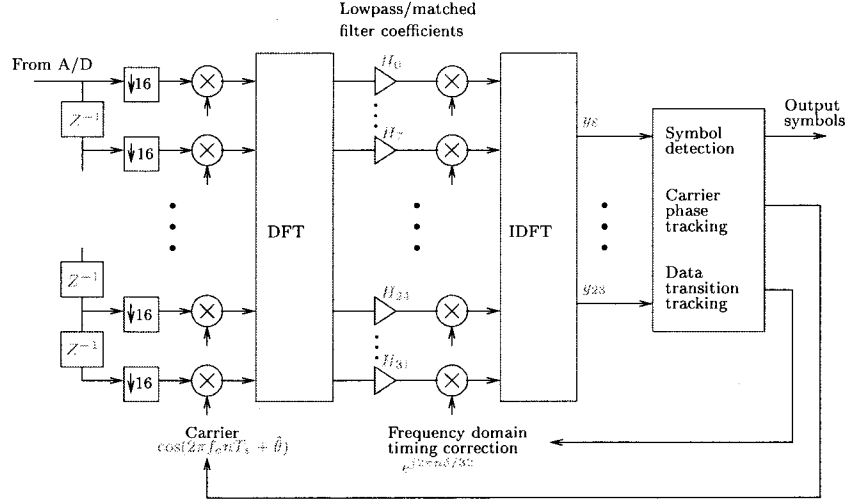
Figure 1: Frequency domain parallel receiver (APRX)

to prevent aliasing following A/D conversion. The filtered analog signal is then bandpass sampled at rate $f_s = 4W$, where W is the transmitted data rate. Note that $f_s = 4W$ is the Nyquist rate for bandpass sampling, and that the IF frequency must satisfy $f_c^{IF} = (2k + 1)W$, for some integer k, in order to avoid aliasing. The sampled IF signal is then digitally mixed with a copy of the IF carrier, the double frequency terms produced by the mixing are rejected by a lowpass filter, and the resulting baseband signal is match filtered, yielding the estimated symbol sequence that may be used by the channel decoder to make bit decisions. By choosing to perform A/D conversion at bandpass rather than at baseband, the carrier phase recovery loop is closed in the digital domain, in keeping with our goal of producing a low cost, flexible, all-digital implementation of receiver functions. The advantages of bandpass sampling over baseband sampling for space applications are discussed in [3].

Matched Filtering

The APRX architecture is based upon implementation of the lowpass and matched filters in the frequency domain via the DFT. Once the noisy IF signal has been filtered and sampled to yield a digital signal with four samples per symbol, the digital signal is split into $2M$ parallel paths, decimated by M, and passed through a digital mixer bank equal in frequency and phase to that of the sampled IF carrier. Adjustments to the carrier phase are provided by the carrier phase tracking loop. The DFT of the $2M$ data points is then calculated and multiplied by the DFT of the matched filter. Lowpass filtering in order to reject double frequency terms from mixing is performed by zeroing out the middle M components in the frequency domain, which correspond to the high frequency terms. Finally, the inverse DFT is performed, and the middle M parallel outputs are used for detection, tracking, etc. This process is repeated once every M A/D clock cycles. In this manner, the processing rate is reduced from f_s to f_s/M. Note that the processing rate for this architecture is not limited by the minimum sampling rate.

The APRX implementation is shown in Figure 1. We let $M = 16$, resulting in 32 parallel signal paths and four symbols output per 16 A/D clock cycles. The 16 points at the output of the IDFT are 16 samples of the convolution integral of the input sequence with the matched filter impulse response function. Among these 16 samples are four peaks that correspond to the matched filter outputs of four symbols. Figure 1 also shows implementation of symbol timing correction, which is discussed later in this paper. There are a few other points to note here. First of all, multiplication of two DFT sequences corresponds to circular convolution in the time domain, and the inverse DFT of this product contains aliased linear convolution values. By parallelizing the input sequence into 32 paths, but decimating only by 16, an overlap is provided so that all of the linear convolution values may be calculated by the overlap and save method. Secondly, by lowpass filtering in the frequency domain via zeroing of high frequency components, we are limited by the resolution of the DFT. This does not appear to pose a problem, however, and
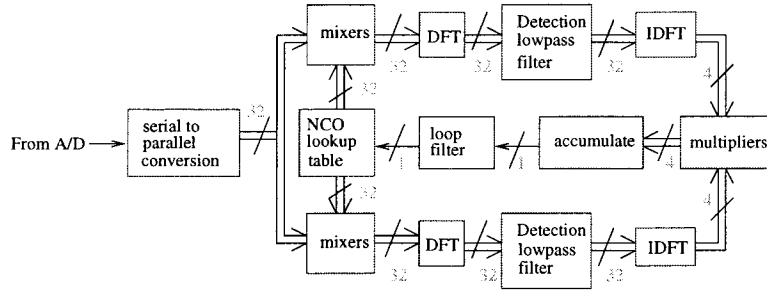
Figure 2: Costas loop for carrier phase tracking.

simulation indicates little or no loss due to this implementation. Finally, we note that the frequency domain matched filter is designed by first designing a time-domain filter matched to the transmitted pulse shape and zero-padding it to length 32, followed by taking the 32-point DFT of the resulting sequence. This yields a frequency domain matched filter whose coefficients are programmed into the detection filter as the H_i values shown in Figure 1.

Carrier Phase Tracking Loop

Carrier phase estimation and tracking is performed in the APRX in a standard fashion, using a Costas loop designed for QPSK signals [4], shown in Figure 2. The double lines represent parallel signal paths. At the output of the IDFT's, only the four pins containing the peaks of the matched filter operation on four symbols are used for phase detection. The inphase and the quadrature components of the parallel arm filters are multiplied to give the phase error, which may be accumulated and then filtered with an IIR filter. This is input to the numerically controlled oscillator, which generates the phase reference used to downconvert the IF signal to baseband (in parallel). The design and analysis of the Costas loop, including specification of loop filter and bandwidth, update rate, etc., follows well developed methodology [4].

Symbol Timing Recovery Loop

In order to implement detection filtering of the baseband signal, the data symbol boundaries need to be known. In a serial digital receiver, an accurate estimate of the symbol phase is needed to adjust the symbol clock so that the matched filter operation is performed on the samples that correspond to the current symbol. For NRZ data, one method of deriving the symbol phase is to use the data-transition tracking loop (DTTL) [4, 5]. In the DTTL, a symbol timing error signal is estimated by summing across a symbol transition in order to measure the deviation from zero. The resulting signal is used to control the numerically controlled oscillator which clocks the sum and dump matched filter interval. There is an inherently finite resolution to the digital DTTL when implemented in this manner due to the fact that symbol phase errors can only be corrected to the extent that samples may be included or excluded from the current symbol, so there is a range of undetectable phase errors.

In the APRX, the peak outputs of the symbol integrators are found as specific pins in the block output of the inverse DFT block of Figure 1. One possible implementation of the DTTL in the APRX would involve calculating the timing error signal from these output pins and using the filtered result to control a commutator that closes the loop by deciding which output pins from the inverse DFT correspond to the correct integrator values. A more natural implementation of the DTTL in the APRX follows from utilizing the frequency domain structure. This implementation is shown in Figure 3. Noting that a time delay corresponds to a phase shift in the frequency domain, we may correct the timing by inserting phase correctors after performing the matched filtering in the frequency domain. This phase correction will have the effect of shifting the desired in-phase and midphase integrator values to a fixed set of selected pins at the output of the inverse DFT. The frequency domain DTTL (FDTTL) is desirable from an implementation standpoint because the required output lines from the inverse DFT are fixed and a commutator routing switch is not needed. More importantly, frequency domain phase correction allows us to effectively solve the problem caused by A/D sampling offset.

In a digital receiver for rectangular NRZ pulses, once there is perfect symbol synchronization, an ideal matched filter detects the kth rectangular pulse data symbol by summing over the samples that are present within the boundaries of the kth symbol. With finite bandwidth causing a distortion in pulse shape, the value of the time offset
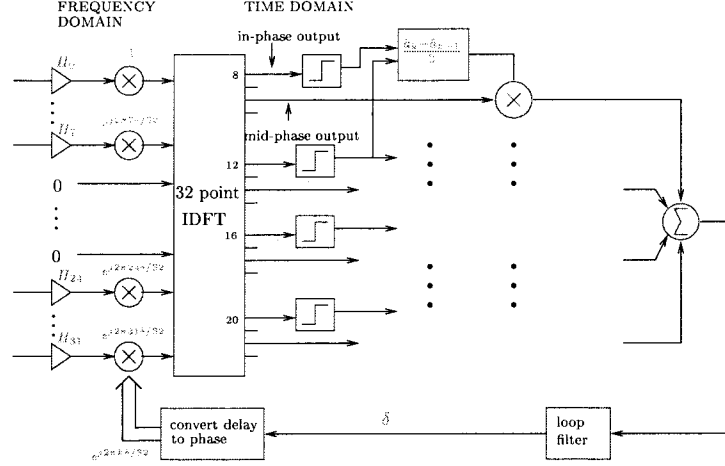
Figure 3: Implementation of parallel DTTL in frequency domain.

of the first symbol sample with respect to the beginning of the pulse will affect the amplitudes of the symbol samples, thereby affecting the output symbol SNR when the samples are summed. The resulting variation in output symbol SNR (and error probability) is more pronounced when there are few samples per symbol. It has been found through simulation that changing the sampling offset from best case to worst case causes a loss of 0.8 to 1.0 dB for rectangular NRZ pulses. This loss is even greater for spectrally-efficient modulation schemes such as SRRC and FQPSK. Two possible remedies for alleviating this loss have been suggested in the past. One is to synchronize the sampling clock with the symbol clock so that the sampling offset is made optimal. This is not desirable if the ultra-stable clock used to synchronize the sampling clock is needed for ranging applications and should not be manipulated, but even otherwise, it is not currently feasible to manipulate the sampling clock when very high data rates are received. A second solution is to use a weighted integrate-and-dump detection filter in which the minimum mean squared error criterion is used to derive coefficients for the detection filter. This equalization method leads a time varying detection filter that changes with the symbol phase output from the symbol synchronization loop.

The solution to dealing with the sampling offset problem arises quite naturally when the frequency domain architecture of the APRX is used. In Figure 3, the phase correction $e^{2\pi k\delta/32}$ that is applied to each frequency domain component k adjusts not only for the integer number of samples that the symbols are delayed by, but also for the fractional number of samples, which corresponds to the sampling offset. In other words, multiplying the N-point discrete Fourier transform of a sequence by $e^{2\pi k\delta/N}$ is equivalent to sampling a delayed version of the continuous time signal. It is shown in [2] that this process drives the symbol timing towards the best case sampling offset situation for NRZ rectangular pulse data. Simulation results indicate that the frequency domain DTTL in the APRX is quite effective for SRRC and FQPSK pulses as well.

SIGNAL MODEL

The continuous time model of the received pulse-shaped QPSK signal is given by

$$r(t) = \sum_{n=-\infty}^{\infty} [a_n p(t - nT_s) \cos(\omega_c t + \theta) + b_n p(t - nT_s - T_s/2) \sin(\omega_c t + \theta)] + n(t) \qquad (1)$$

where $\{a_n\}$ and $\{b_n\}$ are the in-phase and quadrature ± 1 data symbols, $p(t)$ is the transmitted pulse shape, T_s is the symbol duration, ω_c is the carrier frequency, and θ is the received carrier phase. The noise process $n(t)$ is the usual additive white Gaussian noise with two-sided power spectral density $N_0/2$.

For SRRC pulses,

$$p(t) = \frac{4\alpha \left[\cos \left((1 + \alpha)\pi t/T_s \right) + \frac{\sin((1-\alpha)\pi t/T_s)}{4\alpha t/T_s} \right]}{\pi \sqrt{T_s} \left[(4\alpha t/T_s)^2 - 1 \right]}, \qquad (2)$$
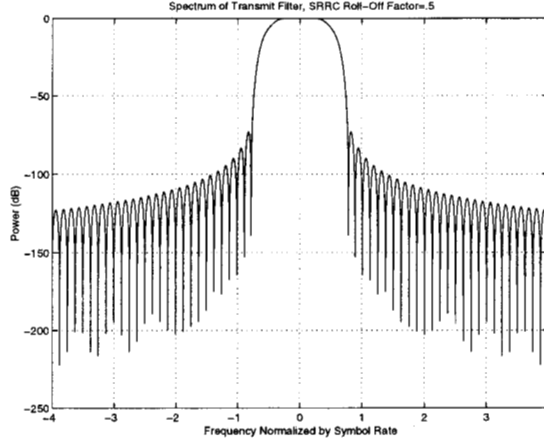
Figure 4: Spectrum of SRRC transmit filter.

with frequency response

$$
P(f) = \begin{cases} \sqrt{T_s} & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \left\{ \frac{T_s}{2} \left(1 + \cos\left[\frac{\pi T_s}{\alpha} \left(|f| - \frac{1-\alpha}{2T_s} \right) \right] \right) \right\}^{1/2} & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0 & |f| > \frac{1+\alpha}{2T_s}. \end{cases} \tag{3}
$$

The ideal transmitter-receiver filter pair consists of two identical SRRC filters with infinite order (infinite time duration) [6]. In practice, the length of the filters must be truncated for implementation purposes. Truncation of the transmit and receiver filters causes energy loss and ISI distortion. In practice, SRRC transmit filters spanning many symbols in length are possible to implement with very small losses on the order of tenths of a decibel. For APRX simulation purposes, a high order 32-tap SRRC transmit filter with roll-off factor of 0.5 whose impulse response spans eight symbols was used to filter the data at the transmitter. The frequency response of this filter is illustrated in Figure 4. Due to the limitation on the implementable size of the frequency domain detection filter in the APRX, the detection filter is truncated to 16 taps, or four symbols. After zero padding to length 32, the 32-point DFT is taken to obtain the coefficients for the frequency domain detection filter.
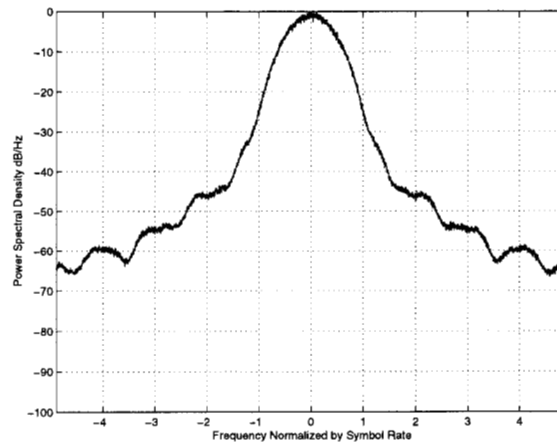

Figure 5: Power spectral density of FQPSK signal.

The FQPSK modulation format has been described in [7] and [8]. It is based upon defining sixteen waveforms over the interval $-T_s/2 \le t \le T_s/2$ whose occurrences in the in-phase and quadrature symbol sequences depend
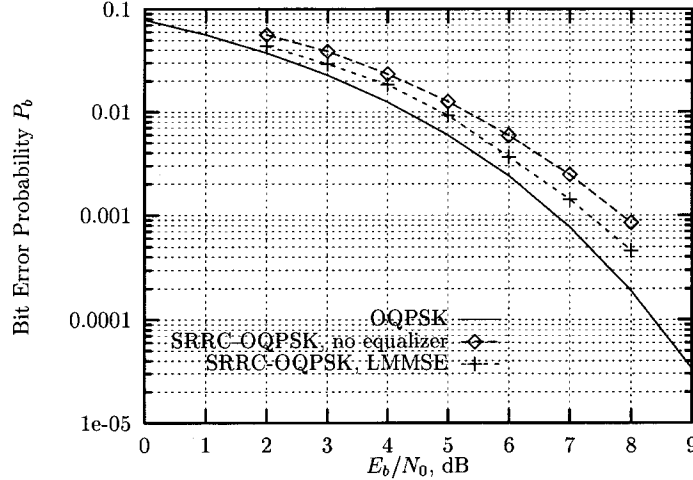
Figure 6: Bit error rate results for SRRC-shaped OQPSK, with no equalizer, and with linear minimum mean squared error equalizer.

upon previous data transitions in both of the channels. The specifics of the symbol mappings are given in [9]. In [9], it was shown that FQPSK could be interpreted as a form of trellis-coded modulation in which a 16-state trellis code takes two binary inputs and outputs in-phase and quadrature waveforms from a set of sixteen pulse shapes. Through this interpretation, it is clear that the maximum likelihood receiver structure for FQPSK consists of a 16-state Viterbi equalizer. The baseline APRX receiver structure, however, is memoryless, i. e., it performs symbol-by-symbol detection, resulting in some performance loss with respect to optimal Viterbi detection.

In the simulations performed here, the transmitted FQPSK waveforms $\{s_i(t) : 0 \leq i \leq 15\}$ were modeled in discrete time with 64 samples per symbol duration. The power spectral density of the baseband FQPSK signal is given in Figure 5.

PERFORMANCE RESULTS

The ideal receiver for SRRC-shaped pulses, using infinite order SRRC transmission and detection filters, yields the same uncoded bit error probability as rectangular OQPSK, $P_b = Q(\sqrt{2E_b/N_0})$. Monte Carlo simulations were conducted using Signal Processing Workstation (SPW) software to determine APRX bit error rate performance. These floating point simulations were run using the full functionality of the APRX, with carrier phase tracking and symbol timing recovery as well as bit detection. Error-control coding is not used in these simulations, as this function is not part of the current APRX design. As shown in Figure 6, the truncated 16-tap detection filter causes about 1.1 dB of loss in performance compared to ideal reception. However, use of a simple linear minimum mean squared error (LMMSE) 8-tap equalizer (spanning 8 symbols) at the back end of the receiver (after symbol detection) brings the bit error rate to within about 0.5 dB of ideal performance.

As mentioned earlier, performance analysis for FQPSK modulation can be obtained in a straightforward manner through the trellis-coding interpretation. The maximum likelihood receiver structure is shown in Figure 7, utilizing a 16-state Viterbi algorithm whose branch metrics are formed from correlations of the in-phase and quadrature components of the received signal with each of the waveforms $\{s_i(t) : 0 \leq i \leq 15\}$ ($s_8(t)$ through $s_{15}(t)$ are the negatives of $s_0(t)$ through $s_7(t)$).

In [9], it was shown that the minimum squared Euclidean distance between pairs of paths in the FQPSK trellis is $d^2_{min} = 1.552T_s$. The average symbol energy for the FQPSK constellation is $\bar{E}_s = 2\bar{E}_b = 0.9946T_s$. As the asymptotic symbol error performance for the trellis code is $Q\left(\sqrt{\left(\frac{d^2_{min}}{2E_b}\right)\frac{\bar{E}_b}{N_0}}\right)$, and the bit error rate is approximately half of
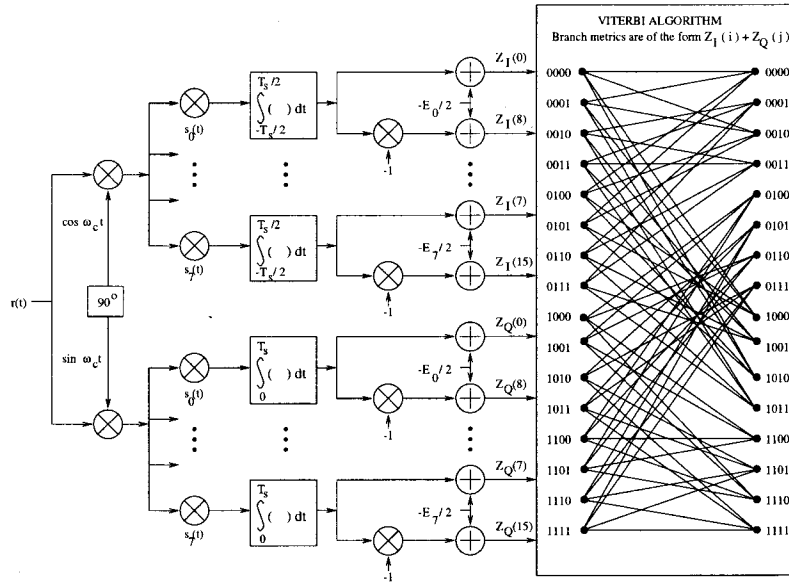
Figure 7: Maximum likelihood receiver structure for FQPSK.

the symbol error rate, the asymptotic bit error rate for maximum likelihood detection of FQPSK is

$$P_b(\text{FQPSK}) \approx \frac{1}{2}Q\left(\sqrt{\frac{1.56\bar{E}_b}{N_0}}\right). \tag{4}$$

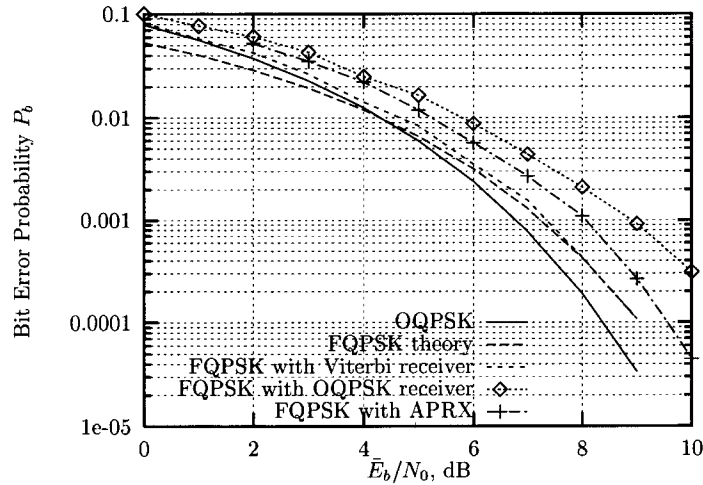This is 1.07 dB worse than ideal OQPSK performance.

Figure 8: Bit error rate results for FQPSK

Suboptimal symbol-by-symbol detection of FQPSK provides a much simpler receiver structure than that shown in Figure 7. For example, the standard integrate-and-dump detector optimal for OQPSK may be applied to FQPSK with an appropriate delay. In the APRX, the detection filter is improved by using an "average" matched filter obtained by experimentally averaging over various combinations of FQPSK waveform sequences. This filter is then implemented in the frequency domain by zero-padding and taking the DFT as described earlier. In Figure 8, bit

error rate curves for various receiver structures are shown. The ideal OQPSK curve is used as a baseline, and is shown along with the asymptotic approximation for Viterbi decoding performance, as well as simulated bit error rates for the Viterbi receiver, conventional OQPSK receiver, and APRX. From this plot, we see that the simulated Viterbi decoding performance converges with the theoretical asymptotic curve of equation (4), and is about 0.7 dB worse than OQPSK in this SNR range. We also see that using the OQPSK symbol-by-symbol integrate-and-dump detector for FQPSK results in an additional 1.7 dB or so of loss. On the other hand, the APRX with the empirical "average" symbol-by-symbol matched filter is within 1 dB away from maximum likelihood performance.

CONCLUSIONS

It has been demonstrated that the advanced parallel digital receiver (APRX) can be used to demodulate SRRC pulse-shaped OQPSK and FQPSK modulation formats with success. For SRRC, the best detection filter that can be implemented in the current APRX design yields poor performance due to the ISI distortion introduced by this filter, but when an eight tap MMSE equalizer is used to reduce ISI distortion on the baseband symbols at the output of the APRX, these losses can be recovered. Performance curves for several receiver structures, including the maximum likelihood Viterbi decoder, were presented for FQPSK, and it was shown that the flexible frequency domain matched filter in the APRX gains at least 0.7 dB in performance over use of the standard integrate-and-dump OQPSK receiver structure.

References

[1] R. Sadr, P. P. Vaidyanathan, D. Raphaeli, and S. Hinedi, "Parallel digital modem using multirate digital filter banks," JPL Publication 94-20, August 1994.

[2] M. Srinivasan, C. Chen, A. Gray, and G. Grebowsky, "All-digital high data rate parallel receiver," *JPL TDA Progress Report*, vol. 42-131, October-December 1997.

[3] R. Sadr and M. Shahshahani, "On sampling band-pass signals," *JPL TDA Progress Report*, vol. 42-96, October-December 1988.

[4] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*. New York: Dover Publications, 1973.

[5] W. C. Lindsey and T. O. Anderson, "Digital-data transition tracking loop," in *International Telemetry Conference*, (Los Angeles, California), pp. 259–271, October 1968.

[6] E. A. Lee and D. G. Messerschmitt, *Digital Communications*. Norwell, Massachusetts: Kluwer Academic Publishers, 1994.

[7] K. F. et al. U. S. Patent 4,567k,602; 4,339,724; 4,644,565; 5,491,457.

[8] K. Feher, *Wireless Digital Communications: Modulation and Spread Spectrum Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[9] M. Simon and T.-Y. Yan, "Performance evaluation and interpretation of unfiltered feher-patented quadrature-phase-shift keying (fqpsk)," *JPL TMO Progress Report*, May 1999.